AI LAB 1

1) Tic Tac Toe bot using python

```
import random
board = [[' ' for _ in range(3)] for i in range(3)]
def game_over():
    for i in range(3):
        if board[0][i] == board[1][i] and board[1][i] == board[2][i] and board[0][i] != ' ':
            return True, board[0][i]
    for i in range(3):
        if board[i][0] == board[i][1] and board[i][1] == board[i][2] and board[i][0] != ' ':
            return True, board[i][0]
    if board[0][0] == board[1][1] and board[1][1] == board[2][2] and board[0][0] != ' ':
        return True, board[0][0]
    if board[0][2] == board[1][1] and board[1][1] == board[2][0] and board[1][1] != ' ':
        return True, board[1][1]
    return False, ' '
def display_board():
    print(f" {board[0][0]} | {board[0][1]} | {board[0][2]}")
    print("--  --  --")
    print(f" {board[1][0]} | {board[1][1]} | {board[1][2]}")
    print("--  --  --")
    print(f" {board[2][0]} | {board[2][1]} | {board[2][2]}")
def num_generator():
    r = random.randint(0,2)
    c = random.randint(0,2)
    return r,c
def take_cell():
    print("Enter the row and column of cell:")
    r = int(input())
    c = int(input())
    return r,c
counter = 0
display_board()
while True:
    r1, c1 = take_cell()
    while board[r1-1][c1-1] != ' ':
        print("Cell already occupied")
        if r1<1 or r1>3 or c1<1 or c1>3:
            print("Invalid cell")
        r1, c1 = take_cell()
    board[r1-1][c1-1] = 'X'
```

```
counter += 1
b, c = game_over()
display_board()
if b:
    print(f"{c} won the game.")
    break
r2, c2 = num_generator()
while board[r2][c2] != ' ':
    r2, c2 = num_generator()
print("Computer played")
board[r2][c2] = 'O'
counter+=1
display_board()
b, c = game_over()
if b:
    print(f"{c} won the game.")
    break
if counter == 9:
    print("Game draw")
    break
```

OUTPUT

```
   |   |
--  --  --
   |   |
--  --  --
   |   |
Enter the row and column of cell:
 2
 2
   |   |
--  --  --
   | X |
--  --  --
   |   |
Computer played
   |   |
--  --  --
   | X |
--  --  --
   |   | O
Enter the row and column of cell:
 3
 1
   |   |
--  --  --
   | X |
--  --  --
 X |   | O
```

```
Computer played
   |   | O
--  --  --
   | X |
--  --  --
 X |   | O
Enter the row and column of cell:
 1
 1
 X |   | O
--  --  --
   | X |
--  --  --
 X |   | O
Computer played
 X |   | O
--  --  --
   | X | O
--  --  --
 X |   | O
O won the game.
```

ALGORITHM

LAB -2

Q) Implement Tic Tac Toe game

→ Algorithm:

```
board = [[' ',' ',' '],[' ',' ',' '],[' ',' ',' ']]
game_over():
    for i <= 0 to 2:
        if board[0][i] = board[1][i] = board[2][i]
            and board[0][i] != ' ':
            return True, board[0][i]
    for i <= 0 to 2:
        if board[i][0] = board[i][1] = board[i][2]
            and board[i][0] != ' ':
            return True, board[i][0]
    if board[0][0] = board[1][1] = board[2][2] != ' ':
        return True, board[0][0]
    if board[0][2] = board[1][1] = board[2][0] != ' ':
        return True, board[1][1]
    return False, ' '
num_gen():
    r = ~~ran gen~~ random.randint(0, 2)
    c = random.randint(0, 2)
    return r, c
take_cell():
    r = int(input())
    c = int(input())
    return r, c
counter = 0
display_board()
while True:
    r1, c1 = take_cell()
    while board[r1-1][c1-1] != ' ':
        r1, c1 = take_cell()
```

```
board[r1-1][c1-1] = 'x'
counter += 1
display_board()
r2, c2 = num_gen()
while board[r2][c2] != ' ':
    r2, c2 = num_gen()
print("computer played!")
board[r2][c2] = 'o'
counter += 1
display_board()
b, c = game_over()
if b:
    print(f"{c} won")
    break
If counter == 9:
    print("Game draw")
    break
```

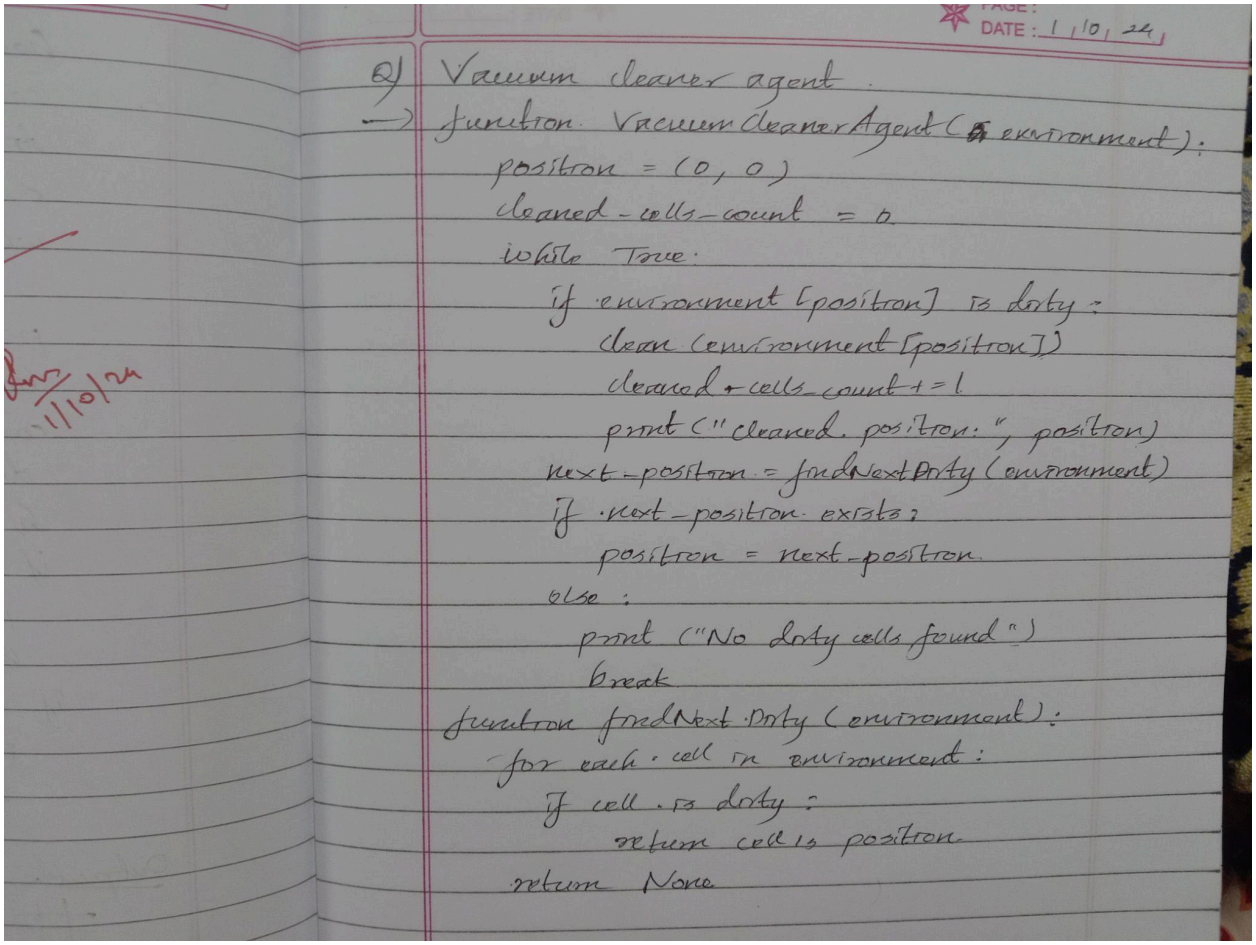
2) Vacuum Cleaner agent

```
import random
l=[random.choice([0,1]),random.choice([0,1])]
def check(i):
    if l[i]==0:
        l[i]=1
        print(f"Cleaned Room {i}")
    print(f"Moved to Room {(i+1)%2}")
    return (i+1)%2
i=random.choice([0,1])
print(f"{i} is the start index")
print("0 is dirty and 1 is clean")
print(f"{l} is the initial state of room")
while sum(l)!=2:
    i=check(i)
    if l[(i+1)%2]==1:
        l[(i+1)%2]=random.choice([0,1])
        if l[(i+1)%2]==0:
            print(f"Room {(i+1)%2} got dirty")
    print(f"{l} is current state of rooms")
print("Rooms are clean")
```

OUTPUT

```
1 is the start index
0 is dirty and 1 is clean
[0, 1] is the initial state of room
Moved to Room 0
[0, 1] is current state of rooms
Cleaned Room 0
Moved to Room 1
[1, 1] is current state of rooms
Rooms are clean
```

ALGORITHM



DATE : 1/10/24

Q) Vacuum cleaner agent.

→ function VacuumCleanerAgent(environment):
    position = (0, 0)
    cleaned_cells_count = 0
    while True:
        if environment[position] is dirty:
            clean(environment[position])
            cleaned_cells_count += 1
            print("cleaned. position:", position)
        next_position = findNextDirty(environment)
        if next_position exists:
            position = next_position
        else:
            print("No dirty cells found")
            break
function findNextDirty(environment):
    for each cell in environment:
        if cell is dirty:
            return cell's position
    return None